Omerta: A Trust-Based Alternative to Blockchain Consensus for Decentralized Compute Markets

Participation Verification Through Subjective Trust and Automated Monetary Policy

Technical Whitepaper

January 2026

Contents

Abstract

Decentralized compute sharing faces a fundamental challenge: how can strangers cooperate without a trusted central authority? Blockchain-based systems address this through proof-of-work or proof-of-stake consensus, achieving Byzantine fault tolerance at the cost of significant resource overhead and limited throughput. We present Omerta, a practical implementation that synthesizes decades of research on trust, reputation, and mechanism design into a working system for decentralized compute markets.

Omerta builds on established foundations—EigenTrust's iterative aggregation [3], FIRE's multi-source trust integration [35], Subjective Logic's uncertainty formalism [36], and transaction cost economics [34]—while making specific design choices suited to compute markets. The key architectural decision is computing trust *locally* relative to each observer, rather than maintaining global scores. This differs from EigenTrust's global PageRank-style computation and enables natural scaling without global consensus overhead.

This paper presents: (1) a trust model derived from verified transactions rather than subjective ratings, building on insights from feedback mechanism design [18, 19]; (2) local trust computation with path decay, extending graph-based trust propagation [3, 37]; (3) automated monetary policy that adjusts parameters in response to detected threats—a novel integration not present in prior reputation systems; (4) economic analysis demonstrating when unreliable home compute creates genuine value, informed by Budish's analysis of blockchain attack economics [38]; and (5) double-spend resolution where currency "weight" scales with network performance, providing a practical alternative to global consensus for bilateral transactions.

We draw on the observation that human societies have always traded privacy for trust—villages had high trust precisely because everyone knew everyone's business. Omerta recreates this visibility at global scale through on-chain transparency. Unlike villages with their arbitrary social punishment, we aim to maximize freedom within the trust constraint: only provably anti-social behavior (failed deliveries, double-spends) affects trust scores, and all mechanisms are documented for scrutiny.

We argue that implementing fair trust systems at scale was computationally intractable until machine intelligence provided the reasoning capacity to model behavior, tune parameters, and explain decisions. This paper itself was developed through human-AI collaboration, demonstrating the thesis: AI both demands the compute that systems like Omerta could provide and enables the trust mechanisms that make such systems work.

Omerta is not presented as theoretically superior to prior work, but as a practical synthesis bringing established ideas into implementation. Where prior work offers better solutions, we aim to adopt them. The contribution is the integration itself—a working system rather than isolated mechanisms.

1. Introduction

The vision of decentralized computing—where anyone can contribute resources and anyone can consume them, without intermediaries extracting rents—has motivated decades of research. The core challenge remains constant: how do strangers cooperate when each has incentive to defect?

1.0 A Brief History of Trust Systems

The question of computational trust saw intensive research in the early 2000s, driven by the rise of peer-to-peer file sharing networks and online marketplaces. EigenTrust [3] adapted PageRank to compute global reputation scores in P2P networks. PeerTrust [4] incorporated transaction context and feedback credibility. FIRE [35] integrated multiple information sources—direct experience, role-based trust, witness reports, and certified reputation. Subjective Logic [36] provided mathematical foundations for reasoning under trust uncertainty. PowerTrust [37] leveraged power-law distributions in feedback patterns for faster convergence.

This body of work established core insights that remain valid: trust propagates through networks with decay; local computation can substitute for global consensus; time and history provide unforgeable credentials; and detection of manipulation patterns enables defensive responses.

Then came Bitcoin (2008) and blockchain, which appeared to solve the trust problem through cryptographic consensus. Research attention shifted. Why model trust computationally when proof-of-work could enforce cooperation mathematically? The reputation systems literature quieted.

A decade later, we understand blockchain's limitations more clearly. Budish [38] demonstrated that blockchain security has inherent economic limits—the recurring costs of running a secure blockchain must be large relative to the value at stake, making it expensive for high-value applications. Proof-of-stake faces similar constraints [39]. Existing decentralized compute networks built on blockchain (Golem [15], iExec [16]) have struggled with adoption despite years of operation. The costs of global consensus may exceed what many applications require.

Meanwhile, machine intelligence has advanced dramatically. Tasks that seemed intractable—modeling complex behavior, tuning parameters across high-dimensional spaces, generating explanations for decisions—are now feasible. This creates an opportunity: **the trust systems research of 2000-2010 may be ready for practical implementation, enabled by AI capabilities that didn't exist when the theory was developed.**

Omerta represents an attempt at this synthesis. We return to the trust-based approaches developed before blockchain's dominance, informed by what we've learned since, and enabled by machine intelligence to handle the complexity that made pure implementation difficult.

1.1 Three Paths

Traditional approaches fall into two categories. **Trusted intermediary** models, exemplified by cloud computing providers, centralize authority in organizations that can enforce contracts and punish misbehavior. This works but introduces single points of failure, censorship risk, and rent extraction. **Blockchain-based** models, exemplified by Ethereum and its descendants, replace trusted intermediaries with cryptographic consensus protocols that theoretically eliminate the need for trust. This also works but imposes significant costs: massive energy expenditure (proof-of-work), capital lockup requirements (proof-of-stake), limited transaction throughput, and delayed finality.

We propose a third path—or rather, we return to one that was overshadowed. Omerta is a trust-based distributed compute network that neither centralizes authority nor attempts to eliminate trust. Instead, it makes trust *subjective*, *local*, and *earned*—computed by each participant from their own position in the network, based on verifiable on-chain data accumulated over time. This approach builds directly on EigenTrust, FIRE, and related work, while making specific adaptations for compute markets and leveraging machine intelligence for implementation.

1.2 The Trust Spectrum

Trustlessness is not binary—it exists on a spectrum. Proof-of-work and proof-of-stake mechanisms genuinely increase trustlessness compared to centralized alternatives. They represent real achievements in distributed systems research. However, historical episodes demonstrate that no system achieves absolute trustlessness:

The DAO Hack (2016): An attacker exploited a smart contract vulnerability to drain $60 million from The DAO. The Ethereum community responded with a hard fork that reversed the theft—creating Ethereum (rolled back) and Ethereum Classic (preserved the "immutable" history). The community chose social consensus over mechanical execution.

Bitcoin Value Overflow (2010): A bug created 184 billion bitcoins out of thin air. Developers and node operators coordinated to deploy a fix and roll back the chain. Human judgment overrode the protocol when stakes were high enough.

Exchange Coordination: When $40 million was stolen from Binance in 2019, the company seriously considered coordinating a Bitcoin rollback before deciding against it. The option existed—revealing that social coordination remains available when needed.

These episodes do not invalidate blockchain achievements. Rather, they reveal that we operate on a spectrum from full trust (centralized authority) to reduced trust (cryptographic consensus) to some irreducible social layer that emerges under sufficient pressure. No practical system reaches the zero-trust endpoint.

The question becomes: given that we cannot achieve absolute trustlessness anyway, what are we paying for the trustlessness we do achieve? And could we relax our requirements slightly to capture most of the practical benefit at dramatically lower cost?

This is the same reasoning that motivates ephemeral compute over fully homomorphic encryption (FHE). FHE provides the ultimate guarantee: compute on encrypted data without ever decrypting it. No trust in the compute provider required. But FHE imposes 1,000-1,000,000x computational overhead [21], making it impractical for most workloads. Ephemeral compute—where data exists briefly on untrusted hardware with verification and economic penalties—provides weaker guarantees but serves far more use cases at practical cost.

Omerta applies this spectrum thinking to consensus itself. Blockchain consensus mechanisms genuinely reduce trust requirements, but at significant cost: energy expenditure (PoW), capital lockup (PoS), limited throughput, and delayed finality. We ask: for compute markets specifically, can we relax the global consensus requirement while preserving the practical security properties that matter?

Our hypothesis is yes. Compute markets do not require global agreement—they require pairwise trust between specific buyers and sellers. By computing trust locally rather than achieving global

consensus, Omerta aims to capture most of the practical benefit of decentralization at dramatically lower cost, making trustworthy compute sharing accessible to more people.

1.3 Our Contribution

This paper's primary contribution is a **practical synthesis**—bringing established trust system research into implementation for compute markets. We distinguish between mechanisms adapted from prior work and novel contributions:

Adapted from prior work (with modifications):

1. **Local trust computation** extending EigenTrust [3] and path-based approaches [4, 37]. Where EigenTrust computes global scores, Omerta computes trust relative to each observer—a design choice explored in the literature but not widely implemented.
2. **Trust propagation with decay**, a standard technique in graph-based trust systems [3, 35, 37], applied here to transaction histories rather than explicit ratings.
3. **Age-based Sybil resistance**, building on temporal defense mechanisms discussed since Douceur's original Sybil attack paper [6]. We note that this defense has known limitations (Section 8.9).
4. **Cluster detection for Sybil defense**, applying standard anomaly detection techniques [9, 10] to transaction graph analysis.

Novel contributions:

5. **Trust from verified transactions only**. Unlike EigenTrust, FIRE, or PeerTrust which incorporate subjective ratings or witness reports, Omerta derives trust exclusively from on-chain transaction records. This eliminates the attack surface of fake feedback but limits the information available for trust computation.
6. **Automated monetary policy** that adjusts economic parameters (payment curves, transfer burns, detection thresholds) in response to observed network metrics. This integration of trust and monetary mechanisms is not present in prior reputation systems.
7. **Double-spend resolution via currency weight**. We show how trust-based systems can handle double-spending without global consensus by scaling finality requirements to network connectivity—a practical alternative for bilateral transactions.
8. **Application to compute markets**. The specific integration of trust, payment, and verification mechanisms for decentralized compute is novel, though individual components draw on established work.

Framing contributions (not technical novelty):

9. **The "resurgence" thesis**: We argue that trust systems research from 2000-2010 is ready for practical implementation, enabled by machine intelligence capabilities that didn't exist then.
10. **AI-assisted design methodology**: This paper itself demonstrates the approach, developed through human-AI collaboration with explicit acknowledgment of that process.

The remainder of this paper is organized as follows. Section 2 reviews related work, explicitly acknowledging the foundations we build on. Section 3 presents system architecture. Section 4

details the trust model. Section 5 describes economic mechanisms. Section 6 analyzes attack vectors and defenses. Section 7 presents simulation results. Section 8 discusses the trust-cost spectrum, limitations, and methodological notes on AI-assisted design. Section 9 concludes.

2. Related Work

2.1 Reputation Systems

The challenge of establishing trust among strangers online has motivated extensive research on reputation systems. Resnick et al. [1] identified the core requirements: long-lived identities, captured feedback, and feedback-guided decisions. The eBay feedback mechanism demonstrated these principles at scale, though its binary ratings created opportunities for manipulation [2].

More sophisticated approaches emerged from peer-to-peer networks in the early 2000s:

EigenTrust [3] computed global trust through iterative aggregation similar to PageRank. Its key insight—that trust can be aggregated through matrix operations—remains foundational. However, it produces global scores rather than observer-relative trust, and relies on explicit transaction ratings.

PeerTrust [4] incorporated transaction context, feedback scope, and community context factors. It recognized that trust depends on more than simple rating counts. Omerta adopts this insight but derives context from transaction records rather than explicit metadata.

FIRE [35] integrated four trust sources: interaction trust (direct experience), role-based trust (position in organization), witness reputation (third-party reports), and certified reputation (references from trustees). This multi-source approach influenced Omerta's design, though we deliberately exclude witness and certified reputation to eliminate subjective input vectors.

Subjective Logic [36] provided mathematical foundations for reasoning under trust uncertainty, modeling opinions as probability distributions with explicit uncertainty parameters. Jøsang's framework for trust transitivity and fusion operations could strengthen Omerta's formal foundations; we note this as future work.

PowerTrust [37] discovered power-law distributions in user feedback patterns and leveraged this for faster convergence through "power nodes." While Omerta doesn't use power nodes, understanding feedback distribution patterns informs our detection of anomalous behavior.

What Omerta borrows: Trust propagation with decay, local computation principles, transaction context sensitivity, and the recognition that different trust components require different handling.

What Omerta changes: We derive trust exclusively from verified on-chain transactions, eliminating subjective ratings entirely. This is a deliberate trade-off: we lose information richness in exchange for removing the fake feedback attack surface. Whether this trade-off is correct depends on the application domain; for compute markets where delivery is verifiable, we believe it is.

2.2 Sybil Resistance

Douceur [6] proved that without a trusted central authority, a single adversary can present arbitrarily many identities indistinguishable from honest participants. This "Sybil attack" undermines any reputation system where influence scales with identity count.

This is a fundamental impossibility result that Omerta does not overcome. We can make Sybil attacks expensive, but not impossible. Honesty requires acknowledging this limitation.

Defenses fall into three categories:

Resource-based: Require each identity to demonstrate control of scarce resources—computational power [7], financial stake [8], or hardware attestation. Effective but expensive, and doesn't prevent well-resourced attackers.

Social-based: Leverage trust graph structure, noting that Sybil identities have sparse connections to honest nodes [9, 10]. SybilGuard [9] and SybilInfer [10] showed that social graph analysis can detect clusters of fake identities. Effective when the social graph reflects real relationships.

Temporal: Require identities to exist over time before gaining influence. This defense, explored in various systems including Freenet's Web of Trust, cannot prevent patient attackers who pre-create identities years in advance.

Omerta employs a hybrid approach: economic penalties (transfer burns), social detection (cluster analysis), and temporal constraints. Identity age—time since on-chain creation—cannot be forged, purchased, or accelerated.

Limitations we acknowledge: A well-resourced attacker who creates thousands of identities today and waits five years will have thousands of mature identities. Omerta's defenses make this expensive in time and capital, but do not make it impossible. We discuss residual attack surfaces in Section 8.9.

2.3 Blockchain Consensus and Its Limits

Bitcoin [7] introduced proof-of-work consensus, achieving Byzantine fault tolerance through computational cost. Subsequent systems explored alternatives: proof-of-stake [8], delegated proof-of-stake [11], practical Byzantine fault tolerance [12], and various hybrid approaches.

All these mechanisms solve the Byzantine Generals Problem: achieving agreement among distributed parties despite malicious actors. This requires $n \geq 3f + 1$ nodes to tolerate f failures, with significant coordination overhead.

Budish [38] demonstrated fundamental economic limits of blockchain security: the recurring flow payments to miners must be large relative to the one-time stock benefits of attacking the system. This makes high-value applications expensive to secure. Gans and Gandal [39] extended this analysis to proof-of-stake, showing similar cost structures manifest as illiquid capital requirements. These analyses suggest blockchain may be over-engineered for applications that don't require global consensus.

Federated approaches occupy a middle ground. Stellar's Federated Byzantine Agreement and Ripple's trust lines allow nodes to choose which other nodes they trust for consensus, rather than trusting the entire network. These systems influenced Omerta's design—the local trust computation is conceptually similar to choosing trusted validators, though Omerta applies this to reputation rather than consensus.

Omerta sidesteps global consensus entirely. Compute markets do not require global agreement—they require pairwise trust between specific buyers and sellers. By computing trust locally, Omerta eliminates the coordination overhead of global agreement while providing the security properties actually needed for compute rental. This is not superior to blockchain for applications requiring global consensus; it is a different trade-off appropriate for different applications.

2.4 Secure Computation Approaches

The ultimate solution to untrusted compute would be **fully homomorphic encryption (FHE)** [21], which enables computation on encrypted data without decryption. FHE provides mathematical guarantees: the compute provider learns nothing about the data. However, current FHE implementations impose 1,000-1,000,000x overhead compared to plaintext computation [22], restricting practical use to narrow applications.

Trusted execution environments (TEEs) like Intel SGX [23] provide hardware-based isolation, but require trusting the hardware manufacturer and have been vulnerable to side-channel attacks [24]. **Secure multi-party computation (MPC)** [25] distributes computation across parties such that no single party learns the inputs, but requires coordination and communication overhead scaling with circuit complexity.

These approaches represent one end of the trust spectrum: maximum guarantees at maximum cost. Omerta explores the opposite trade-off: accepting some trust requirements in exchange for practical performance that serves more use cases.

2.5 Decentralized Computing

Distributed computing projects like BOINC [13], Folding@home [14], and SETI@home demonstrated that volunteers would contribute compute resources for scientific research. Commercial successors like Golem [15] and iExec [16] built on blockchain platforms to enable paid compute sharing.

These systems face common challenges: verifying that claimed work was actually performed, preventing providers from delivering inferior resources, and detecting collusion. Omerta addresses these through continuous verification, trust-based payment splits, and statistical detection of manipulation patterns.

2.6 Computational Economics and Mechanism Design

The design and validation of economic mechanisms increasingly relies on computational methods. Agent-based computational economics [26, 27] provides tools for studying emergent phenomena in complex markets where analytical solutions are intractable [28]. This approach has proven particularly valuable for mechanism design [29], where simulating agent behavior under proposed rules reveals edge cases and failure modes before deployment.

Validation of agent-based models follows established practices [30]: parameter sensitivity analysis, comparison against theoretical predictions where available, and testing under adversarial conditions. These methods inform our simulation methodology in Section 7.

The market design draws on auction theory [31] and matching market literature [32]. The trust propagation model relates to work on reputation mechanism design [33], particularly the challenge of eliciting honest feedback in the presence of moral hazard. The concept that trust mechanism overhead should scale with network uncertainty echoes transaction cost economics [34], which analyzes how institutions emerge to reduce uncertainty in exchange.

Omerta's economic simulations build on these foundations while extending them to a novel domain: trust-based compute markets where machine intelligence both creates demand and enables the trust mechanisms that make supply possible.

3. System Architecture

3.1 Design Principles

Omerta is built on several core principles:

Verifiable Facts, Subjective Trust: The blockchain stores facts (transactions, verification logs, assertions). Trust scores are computed locally by each participant from these facts according to their own criteria.

Identity Age as Credential: Of all possible credentials, only time-on-chain cannot be manufactured. New identities start with nothing and must earn trust through participation.

Uniform Pricing, Trust-Based Splits: All consumers pay the same market rate. Trust determines how payments split between provider and burn. High trust = more to provider. Low trust = more burned.

Earned Write Permission: Only identities above trust thresholds can write to the chain, with conflict resolution through trust-weighted voting. Trust score IS the stake.

3.2 On-Chain Data

The blockchain records five types of data:

Identity Records: Public key, creation timestamp, and signature capabilities. All derived data (age, transaction count, trust scores) is computed from other records.

Transactions: Consumer and provider identities, amounts paid/received/burned, resource specifications, duration, and signatures from both parties.

Verification Logs: Results of resource checks, including verifier identity, claimed vs. measured resources, pass/fail result, and verifier signature.

Trust Assertions: Signed claims by one identity about another, including score, classification (positive or negative), evidence hashes, and reasoning.

Order Book: Bids and asks for compute resources, enabling price discovery through market mechanisms.

3.3 Market Structure

Omerta implements an on-chain order book for price discovery:

Order Types: Bids (consumer wants to buy) and asks (provider wants to sell), with resource specifications, price, duration constraints, and expiration.

Resource Classes: Standardized categories (small_cpu, medium_cpu, gpu_consumer, gpu_datacenter) enabling liquidity aggregation.

Matching Engine: Price-time priority matching. When orders cross, sessions initiate and escrow triggers.

Spot Rate: Volume-weighted average of recent trades, providing simple consumer-facing pricing while preserving price discovery.

Consumers see simple pricing ("8 cores = 0.09 OMC/hr") without needing to understand the underlying market mechanics.

3.4 Session Lifecycle

A compute session proceeds as follows:

1. **Order Placement**: Consumer places bid or provider places ask
2. **Matching**: Orders cross, session initiates
3. **Escrow Lock**: Consumer's payment locked in escrow
4. **VM Configuration**: Provider configures VM with consumer's identity key for access
5. **Compute Execution**: Consumer uses resources
6. **Verification**: Random audits check resource claims
7. **Settlement**: Escrow released based on outcome and trust scores

Either party can terminate at any time—the market handles quality through consumer exit and provider reliability signals.

4. Trust Model

4.1 Trust Accumulation

Trust accumulates from verified transactions, not subjective ratings:

$$T_{base} = T_{transactions} + T_{assertions}$$

$$T_{transactions} = \sum_i (CREDIT \times resource\_weight_i \times duration_i \times verification\_score_i \times cluster\_weight_i)$$

Transaction-based trust grows with verified compute provision. Each term serves a specific purpose: resource weights normalize across compute types, duration captures commitment extent, verification scores reflect audit outcomes, and cluster weights downweight suspected Sybil transactions.

Assertion-based trust adjusts for reported incidents:

$$T_{assertions} = \sum_i (score_i \times credibility_i \times decay_i)$$

Assertions are signed reports of specific incidents with scores in [-1, 1]. Positive scores (commendations) add trust; negative scores (violations) subtract. Credibility derives from the asserter's own trust, creating recursive dependency resolved through iterative computation.

4.2 Age as Derate Factor

A critical design choice: age should **never add trust**, only remove a penalty from young identities:

$$T_{effective} = T_{base} \times age\_derate$$

$$age\_derate = \min\left(1.0, \frac{identity\_age}{AGE\_MATURITY\_DAYS}\right)$$

New identities start at zero effective trust regardless of transaction volume. This prevents attackers from pre-creating dormant identities that accumulate trust through mere existence. You can only earn trust by participating over time.

4.3 Local Trust Computation

Trust is not global. Each observer computes trust relative to their position in the network:

$$T(subject, observer) = T_{direct} + T_{transitive}$$

$$T_{transitive} = \sum_{intermediary} T(intermediary, observer) \times T(subject, intermediary) \times DECAY^{path\_length}$$

Direct trust comes from personal transaction history. Transitive trust propagates through trusted intermediaries with exponential decay per hop.

Why this matters: An attacker cannot build trust in Community A and exploit it in Community B. Observers in B see the attacker's trust discounted by lack of network path. The attacker must build trust directly with each community they wish to exploit—exactly how human trust works.

4.4 Parameterized Infractions

Not all violations are equal. Infraction severity scales with potential network impact:

$$effective\_score = base\_score \times impact\_multiplier \times context\_multiplier$$

Impact scales with transaction value, resources affected, and duration. Context includes repeat offense history. This creates appropriate gray areas—small mistakes don't destroy trust, while large attacks risk proportional penalties.

5. Economic Mechanisms

5.1 Payment Splits

Consumer payments split between provider and cryptographic burn based on provider trust:

$$provider\_share = 1 - \frac{1}{1 + K_{PAYMENT} \times T}$$

As trust increases, provider share approaches 100% asymptotically but never reaches it. New providers with low trust see most of their payment burned; established providers keep most of it.

This creates natural incentives: build trust to keep more of your earnings. No external enforcement needed—the economics handle it.

5.2 Transfer Burns

Transfers between identities are taxed based on the minimum trust of sender and receiver:

$$transfer\_burn\_rate = \frac{1}{1 + K_{TRANSFER} \times \min(T_{sender}, T_{receiver})}$$

Low-trust identities cannot easily transfer coins. This prevents reputation laundering (build trust, exploit, transfer coins to fresh identity) and creates strong incentive to donate compute rather than buy coins.

5.3 Daily Distribution

New coins are minted daily and distributed proportionally to trust scores:

$$daily\_share(i) = \frac{effective\_trust(i)}{\sum_j effective\_trust(j)} \times DAILY\_MINT$$

This creates the core incentive: misbehave and your trust drops, reducing tomorrow's share. The motivation to be honest is simply: keep getting your handout.

5.4 Donation and Negative Bids

Providers can accept negative-price bids from research organizations, burning their own coins for accelerated trust:

Bid Type	Price	Trust Multiplier
Commercial	Positive	1x
Zero donation	Zero	1x
Negative donation	Negative	Up to 4x

This enables bootstrapping: new providers can burn coins to accelerate trust building while providing verified compute to research projects. Trust cannot be purchased without actual work—the burn is additional, not replacement.

6. Attack Analysis and Defenses

6.1 Sybil Attacks

Attack: Create many fake identities to manipulate trust or distribution.

Defenses: - **Age derate**: New identities earn nothing initially - **Cluster detection**: Tightly-connected subgraphs with few external edges are flagged - **Behavioral similarity**: Identities behaving too similarly are downweighted - **Activity requirements**: Must maintain ongoing participation

6.2 Collusion and Trust Inflation

Attack: Colluding parties mutually vouch for each other to inflate trust.

Defenses: - **Graph analysis**: Detect circular trust flows and isolated cliques - **Verification requirements**: Trust requires verified compute, not just assertions - **Statistical anomaly detection**: Burst activity, coordinated timing flagged

6.3 Trust Arbitrage

Attack: Build trust in Community A, exploit in Community B.

Defense: Local trust computation means trust doesn't transfer across network distance. Attackers must build trust directly with each target community.

6.4 Multi-Identity Exploitation

Attack: Maintain multiple identities to hedge risk or enable sacrificial attacks.

Critical Distinction: Some multi-identity strategies must be absolutely prevented; others may be tolerated.

Absolute Protections: - UBI distribution: Malicious behavior in any linked identity reduces combined distribution - Trust from activity: Same work split across N identities yields at most single-identity trust - Accusation credibility: N low-credibility accusations don't sum to high credibility

Tolerated Advantages: - Risk diversification: Legitimate businesses may operate multiple identities - Community separation: Operating in isolated communities without cross-contamination - Recovery via new identity: Starting fresh with appropriate penalties

6.5 Identity-Bound Access

Problem: Traditional credential theft enables exploiting stolen identity's reputation.

Solution: VM access is bound to the consumer's on-chain private key. No key, no access. If you have the private key, you ARE the identity—there is no "stealing," only "being."

7. Simulation Results

We conducted simulation studies testing the automated monetary policy under adversarial conditions. Key findings:

7.1 Attack Scenario Outcomes

Scenario	Final Gini	Cluster Prevalence	Policy Response
Baseline (Honest)	0.783	0.000	Stable
Trust Inflation	0.706	0.250	K_TRANSFER increased
Sybil Explosion	0.641	0.545	ISOLATION_THRESHOLD decreased
Gini Manipulation	0.882	0.000	K_PAYMENT decreased

7.2 Key Finding: Structural vs. Parametric Attacks

The simulations revealed a striking pattern: automated policy adjustments trigger correctly but have limited impact on final outcomes. This suggests that attack effects are primarily structural rather than parameter-dependent.

Implication: Effective attack resistance requires architectural defenses that make attacks structurally infeasible, complemented by policy adjustments for fine-tuning. Parameter tweaking alone is insufficient against determined adversaries.

7.3 Long-Term Stability

Five-year simulations showed stable trust accumulation under honest conditions and recovery between attack waves under adversarial conditions. The core trust mechanisms prove robust to extended adversarial pressure.

7.4 Reliability Market Economics

A critical question for any decentralized compute network: can unreliable home providers coexist with reliable datacenters, or will one displace the other? We simulated provider competition under varying market conditions.

Provider Cost Structures:

Provider Type	Cost/hr	Reliability	Cancellation
Datacenter	\$0.50	99.8%	Never (SLA)
Home Provider	\$0.08	92%	For profit (1.5× threshold)

Home providers have 6× lower costs because they already own hardware (no capex amortization), pay only marginal electricity, and have no facility overhead.

Market Outcomes by Supply/Demand:

Market Condition	Consumer Cost Δ	DC Profit Δ	Compute Δ
Undersupplied (uniform values)	-10%	**-3%**	**+200%**
Undersupplied (mixed values)	-52%	-66%	+200%
Balanced	-91%	-100%	+70%
Oversupplied	-84%	-100%	-8%

Key Findings:

1. **Undersupplied markets show value creation**: When demand exceeds datacenter capacity, home providers serve unmet demand. Total compute delivered increases 200% while datacenter profits remain largely intact (only -3% with uniform consumer values).
2. **Balanced/oversupplied markets show displacement**: When total capacity meets or exceeds demand, home providers' 6× cost advantage allows them to undercut datacenters completely.
3. **Consumer heterogeneity matters**: When consumers have widely varying values per compute hour, middle-market competition intensifies. Datacenters retain only the premium segment.
4. **Reliability tradeoff is real**: In oversupplied markets, total useful compute can decrease (-8%) because home provider unreliability causes more restarts despite lower prices.

Implication for Omerta: The system creates genuine economic value only when **demand exceeds datacenter capacity**. This raises the question: will demand ever sustainably exceed supply?

7.5 The Machine Intelligence Demand Thesis

We argue that machine intelligence fundamentally transforms compute markets into **perpetually undersupplied markets**, making unreliable compute economically valuable for the foreseeable future.

The Traditional View:

Human-driven compute demand is bounded. Businesses have finite workloads, consumers have finite entertainment needs. Markets tend toward equilibrium where supply meets demand. In equilibrium, price competition drives out high-cost providers.

Under this view, home providers would be viable only during transient demand spikes.

The New Reality:

Machine intelligence creates unbounded demand because **machines can always find productive uses for additional compute**. Unlike humans, who run out of tasks to do, AI systems have continuous demand curves:

Task Priority	Human Value	Machine Value	Notes
Frontier research	\$100/hr	\$0/hr	Requires human insight (for now)
Active inference	\$50/hr	\$40/hr	Real-time decision making
Background reasoning	\$20/hr	\$15/hr	Exploring solution spaces
Speculative search	\$5/hr	\$3/hr	Low-priority but positive value
Precomputation	\$1/hr	\$0.50/hr	Preparing for future queries

The key insight: **there is always a next-best task**. A machine that cannot profitably do a \$50/hr task can still generate value doing a \$5/hr task. This creates a demand curve extending to arbitrarily low prices.

Implications:

1. **No "oversupplied" market exists**: Machine demand expands to absorb any available compute
2. **All providers can coexist**: Datacenters serve high-priority tasks, home providers serve the elastic tail
3. **Price floors are set by marginal value, not marginal cost**: Machines bid what tasks are worth

Future Projection:

As machine intelligence improves, the value per compute hour increases at all quality levels:

Era	Demand Characteristic	Market Structure
Human-only	Bounded, tends to equilibrium	Oversupply risk
Human + subhuman AI (today)	Large, finite	Undersupplied
Human + superhuman AI (future)	**Unbounded**	Permanently undersupplied

At the limit, superintelligent systems have unlimited demand for compute of any quality. Any machine cycle has positive value because it can be applied to self-improvement, exploration, or capability expansion.

Conclusion:

The economic viability of Omerta depends on perpetual undersupply. Human demand alone cannot guarantee this. But machine intelligence—which can always find productive uses for marginal compute—transforms the market structure fundamentally. In a world of AI agents, the question is not whether unreliable compute will displace datacenters, but whether we can deploy enough compute of any quality to satisfy exponentially growing machine demand.

7.6 Double-Spend Resolution

Unlike blockchain where double-spending is mathematically prevented by consensus, Omerta's trust-based currency can only detect and penalize double-spends after the fact. This raises questions about economic stability. We simulated five scenarios to validate the design.

7.6.1 Detection Rate

Connectivity	Detection Rate	Avg Time	Network Spread
0.1	100%	0.046s	97.6%
0.5	100%	0.042s	98.0%
1.0	100%	0.042s	98.0%

Finding: In gossip networks, double-spends are always eventually detected because conflicting transactions propagate to common nodes. Connectivity affects detection speed, not completeness.

Important clarification: Detection is not prevention. A 100% detection rate means the double-spend is always discovered—but only *after* the conflicting transactions have propagated. During the detection window, both recipients may believe they have valid payments. The economic defense (trust penalties, Section 7.6.2) makes attacks unprofitable but does not prevent the temporary confusion. For high-value transactions where even temporary double-spend would be harmful, use the "wait for agreement" protocol (Section 7.6.3) which provides prevention through delayed finality.

7.6.2 Economic Stability

The "both keep coins" strategy (accept inflation, penalize attacker) shows:

Detection	Penalty	Inflation	Attacker Profit	Stable?
50%	5x	1.9%	-$985	YES
90%	5x	1.1%	-$1000	YES
99%	1x	4.7%	-$943	YES

Finding: Attackers always lose money because trust penalties outweigh gains. Economy is stable (inflation $< 5\%$) with 5x penalty multiplier even at 50% detection.

7.6.3 Finality Latency

Threshold	Connectivity	Median Latency	Success Rate
50%	0.3	0.14s	100%
70%	0.5	0.14s	100%
90%	0.7	0.14s	100%

Finding: Sub-200ms finality achievable. Higher thresholds don't proportionally increase latency because confirmations arrive in parallel through gossip.

7.6.4 Partition Behavior

During network partitions, double-spends can temporarily succeed (both victims accept). After healing, all conflicts are detected. This creates a "damage window" equal to partition duration.

Mitigation: Use "wait for agreement" protocol for high-value transactions during suspected partition conditions.

7.6.5 Currency Weight Spectrum

Connectivity	Weight	Category
0.9	0.14	Lightest (village-level trust)
0.5	0.34	Light (town-level)
0.1	0.80	Heaviest (needs blockchain bridge)

Conclusion: Currency weight is proportional to network performance. Better connectivity enables lighter trust mechanisms—the digital equivalent of physical proximity enabling village-level trust at global scale.

8. Discussion

8.1 The Trust-Cost Spectrum

Different approaches occupy different positions on the trust-cost spectrum:

Approach	Trust Required	Cost	Practical Scope
Centralized cloud	High (trust provider)	Low	Broad
FHE	None	Extreme (1000x+)	Very narrow
TEE (SGX)	Medium (trust hardware)	Low-Medium	Medium
PoW blockchain	Low	High (energy)	Medium
PoS blockchain	Low-Medium	Medium (capital)	Medium
Omerta	Medium (trust earned)	Low	Broad

The key insight is that blockchain approaches—while genuinely reducing trust requirements—may not occupy the optimal point for compute markets. They pay significant costs (energy, capital, throughput limits) for global consensus that compute markets may not need.

8.2 What Blockchain Achieves

We should be precise about what blockchain consensus mechanisms accomplish:

Genuine achievements: - Coordination without designated coordinator - Resistance to unilateral censorship - Auditable history without trusted record-keeper - Credible monetary policy without central bank

These are real and valuable. PoW and PoS represent genuine advances in reducing trust requirements.

What the historical episodes reveal: - The spectrum has no zero-trust endpoint in practice - Social consensus remains available when stakes are high enough - This doesn't invalidate the achievements—it bounds them

8.3 Omerta's Position

Omerta bets that for compute markets specifically, we can relax global consensus while preserving the properties that matter:

Property	Blockchain Approach	Omerta Approach	Trade-off
Double-spend prevention	Global UTXO consensus	Escrow locks before session	Equivalent for sessions
History integrity	PoW/PoS difficulty	Trust-weighted writes	Weaker globally, sufficient locally
Sybil resistance	Computational/capital cost	Time cost (age)	Different attack economics

Property	Blockchain Approach	Omerta Approach	Trade-off
Censorship resistance	Anyone can mine/stake	Anyone above trust threshold	Requires earning entry

The hypothesis is that these trade-offs are acceptable for compute markets, where: - Transactions are bilateral (buyer-seller), not global transfers - Sessions are ephemeral, not permanent state - Verification is possible during execution - Reputation has natural meaning (did you deliver?)

8.4 Analogy to FHE vs. Ephemeral Compute

The relationship between Omerta and blockchain mirrors the relationship between ephemeral compute and FHE:

	Maximum Guarantee	Practical Alternative
Data privacy	FHE (compute on encrypted)	Ephemeral compute (brief exposure + verification)
Consensus	Global Byzantine (PoW/PoS)	Local trust (Omerta)
Cost	1000x+ overhead	Near-native performance
Accessibility	Narrow applications	Broad applicability

In both cases, relaxing the maximum guarantee enables serving far more users at practical cost. The question is whether the relaxed guarantee is sufficient for the use case.

8.5 Making the Social Layer Explicit

Every distributed system ultimately has a social layer. Bitcoin's ledger has been modified by human coordination. Ethereum's code yielded to community override. The question is not whether humans are trusted, but which humans and how.

Omerta makes this explicit rather than hiding it beneath claims of mathematical trustlessness. Trust relationships are tracked on-chain. Incentives align through economics. Bad actors are identified over time. This mirrors how working human institutions operate—imperfect rules made workable through aligned incentives.

8.6 Interoperability Across the Spectrum

Different points on the trust-cost spectrum suit different use cases. A mature ecosystem might include multiple networks that interoperate, with value and workloads flowing to appropriate trust levels.

Use cases by trust level:

Trust Level	Example Use Cases	Why This Level
Maximum (FHE/MPC)	Medical records analysis, financial audits, voting	Data must never be exposed, even briefly

Trust Level	Example Use Cases	Why This Level
High (PoW/PoS blockchain)	Digital currency, smart contracts with large stakes, cross-border settlements	Global consensus needed, high-value permanent state
Medium (Omerta-style)	General compute rental, batch processing, development environments, CI/CD	Bilateral transactions, ephemeral sessions, verification possible
Lower (reputation only)	Content delivery, caching, non-sensitive workloads	Speed matters more than guarantees, easy to verify after the fact

Cross-network flows:

Networks at different trust levels can bridge to each other:

- **Settlement on high-trust chains**: An Omerta-style network could settle periodic summaries to a PoS blockchain, gaining the permanence guarantees of global consensus for aggregate state while handling high-frequency bilateral transactions locally.
- **Escalation for disputes**: Normal compute sessions run on medium-trust infrastructure. Disputed sessions escalate to higher-trust arbitration—perhaps a smart contract on Ethereum that evaluates cryptographic evidence.
- **Sensitive workload isolation**: A pipeline might run preprocessing on Omerta (cheap, fast), then route sensitive computation to FHE or TEE enclaves, then aggregate results back on Omerta.
- **Trust bootstrapping**: New participants could establish initial reputation on a high-trust chain (where Sybil attacks are expensive), then bridge that identity to lower-cost networks.

The vision:

Rather than one network attempting to serve all use cases at one trust level, the ecosystem stratifies. Users and workloads flow to appropriate levels based on their actual security requirements. Most computation doesn't need the guarantees (or costs) of global consensus. Some does. A well-designed ecosystem makes both available and composable.

This is analogous to how the internet layers protocols: TCP provides reliable delivery at some cost, UDP provides speed without guarantees, and applications choose based on their needs. Similarly, a trust-stratified compute ecosystem would let applications choose their trust level, paying only for the guarantees they actually require.

8.7 Scaling Trust: From Villages to Global Networks

The trust-cost tradeoff mirrors how human societies have always operated:

Society Scale	Trust Mechanism	Overhead
Village (50)	Everyone knows everyone; gossip spreads instantly	Minimal

Society Scale	Trust Mechanism	Overhead
Town (5,000)	Reputation networks; friends-of-friends	Low
City (500,000)	Institutions track reputation; courts enforce	Medium
Nation (50M+)	Anonymous transactions; verification required	High
Global	No shared context	Highest

As communities grew beyond the scale where everyone could know everyone, **visibility decreased** and **trust costs increased**. The lightweight mechanisms that worked in villages—verbal agreements, handshakes, reputation by gossip—don't scale to cities. Heavier mechanisms emerged: contracts, courts, banks, regulations.

The core problem: Traditional high-trust solutions required physical proximity and small scale. You could trust your neighbor because you'd see them tomorrow and everyone would know if they cheated you.

What Omerta provides:

1. **Visibility at scale**: On-chain records make transaction history visible to anyone, replicating the "everyone knows" property of villages at global scale.
2. **Gossip with perfect memory**: In villages, reputation spreads by word of mouth but decays over time. Omerta makes reputation computable from permanent records.
3. **Investment and lock-in**: Building trust over time creates "skin in the game." Defection costs accumulated reputation, making honest behavior rational even among strangers.
4. **Graduated entry**: New participants start with nothing, exactly like newcomers to a village. Trust is earned through demonstrated behavior.

The key insight: **network performance is the digital analog of physical proximity**. Nodes that communicate quickly and reliably can use lighter trust mechanisms, just as neighbors who see each other daily can trust more easily than strangers across the world.

Omerta asks: if we provide village-level visibility at global scale, can we use village-weight trust mechanisms for global transactions? The answer appears to be yes—to the extent that network performance allows. Currency weight becomes proportional to the proximity (physical or digital) between participants.

The freedom-trust tradeoff: This visibility comes at a cost. Villages had high trust precisely because they had low freedom—everyone knew your business, you couldn't easily leave, your reputation followed you everywhere. Omerta recreates these properties digitally: transactions are visible, identities are persistent, exit costs are high, and deviation is penalized.

This is not a bug—it is the mechanism by which trust scales. The system explicitly trades freedom for trust. Users who value anonymity, disposable identities, or the ability to "start fresh" should use different systems. Users who value counterparty trust, long-term reputation, and cooperation among strangers may find this tradeoff worthwhile.

The honest framing: Omerta is not a privacy technology. It is an anti-privacy technology that trades surveillance for trust. The design should be chosen knowingly, when that tradeoff serves the application's needs.

Avoiding village pathologies: While drawing on village-level trust mechanisms, Omerta explicitly aims to avoid the well-known pathologies of small communities: arbitrary social punishment, gossip and rumor, in-group favoritism, and hidden power structures. The goal is to **maximize freedom within the trust constraint**:

1. *Only penalize provable misbehavior*: Trust scores decrease only for objectively measurable, demonstrably harmful actions (failed delivery, double-spend attempts). Subjective judgments have no mechanism to affect scores.
2. *Measure in the open*: All data used to compute trust scores is on-chain and visible. No hidden inputs, no secret algorithms.
3. *Keep mechanisms debatable*: Because everything is transparent, the community can debate fairness. Governance adjusts parameters based on observed results.
4. *Preserve freedom for non-harmful behavior*: No constraints on what compute is used for, who you transact with, or how you operate—only on whether you fulfill commitments.

This distinguishes Omerta from both the capricious social control of villages and the opaque algorithmic control of centralized platforms. It aims to be a *fair* surveillance system: comprehensive but explicit, uniform, and challengeable.

Why now: Machine intelligence as enabler

The reason fair, transparent trust systems at scale weren't possible before is computational: **modeling, tracking, and adjusting parameters that sufficiently cover human behavior requires enormous reasoning capacity**.

Villages could be fair because scope was small—a few hundred relationships, elders who remembered everything. Scaling that to millions of participants with complex, evolving behavior was computationally intractable. Machine intelligence changes this:

Challenge	Traditional	With AI
Defining "anti-social"	Static rules, courts	Dynamic models learning edge cases
Detecting misbehavior	Manual review	Continuous automated analysis
Explaining decisions	Impenetrable legalese	Natural language on demand
Adjusting parameters	Years of committee debate	Rapid simulation-validated iteration

Machine intelligence enables: (1) behavioral modeling at scale, distinguishing honest mistakes from malice; (2) continuous parameter tuning through simulation; (3) explanation generation for trust score decisions; (4) adversarial reasoning against novel attacks; (5) governance support with AI-generated outcome analysis.

The relationship is recursive: **machine intelligence both demands the compute that Omerta provides and enables the trust system that makes Omerta work**. AI needs distributed compute; distributed compute needs trust mechanisms; trust mechanisms at scale need AI to operate fairly. This virtuous cycle suggests the timing is not coincidental—the technologies arrive together because each enables the others.

8.8 Methodological Notes: AI-Assisted System Design

This paper as demonstration: This paper and its accompanying simulations were developed with substantial assistance from large language models (specifically Claude). This is not incidental—it is a direct demonstration of the thesis that machine intelligence enables system design that was previously intractable. The economic models, attack analyses, parameter explorations, and simulation code were developed iteratively through human-AI collaboration, with the AI providing rapid prototyping, edge case identification, and documentation while the human provided direction, validation, and domain judgment.

We acknowledge this explicitly because intellectual honesty demands it, and because it illustrates both the strengths and weaknesses of AI-assisted design.

Connection to computational economics literature: The simulation methodology draws on established traditions in agent-based computational economics [26, 27]. Agent-based models have proven valuable for studying emergent phenomena in markets [28] and mechanism design [29], particularly when analytical solutions are intractable. Our approach follows the validation practices recommended by Windrum et al. [30]: parameter sweeps, sensitivity analysis, and comparison against theoretical predictions where available.

The market mechanism design builds on auction theory [31] and matching market literature [32], while the trust propagation model relates to work on network effects in reputation systems [33]. The "currency weight" concept—that trust mechanism overhead should scale with network uncertainty—echoes insights from transaction cost economics [34].

Strengths of AI-assisted analysis:

1. *Rapid iteration*: Parameter spaces that would take months to explore manually can be swept in hours, enabling broader coverage of edge cases.
2. *Consistency*: AI maintains consistent reasoning across long documents, catching contradictions between sections that human authors might miss.
3. *Adversarial thinking*: AI can systematically enumerate attack vectors, reducing the risk of overlooking obvious vulnerabilities.
4. *Documentation*: AI excels at explaining reasoning, making the logic auditable in ways that purely human intuition is not.

Weaknesses and mitigations:

1. *Hallucination risk*: AI may generate plausible-sounding but incorrect analysis. **Mitigation**: All simulation code is executable and produces verifiable outputs. Claims are grounded in simulation results, not AI assertions. Key findings were validated against analytical intuition.
2. *Training data limitations*: AI knowledge has a cutoff date and may miss recent developments. **Mitigation**: Human collaborators review for currency and supplement with recent literature.

3. *Sycophancy bias*: AI may agree with human suggestions rather than pushing back on flawed reasoning. **Mitigation**: Explicitly adversarial prompting ("what's wrong with this?", "how would an attacker exploit this?") and independent verification of key claims.

4. *Lack of true understanding*: AI manipulates symbols without grounded understanding of what they mean. **Mitigation**: Simulation provides empirical grounding. If the model is wrong, the simulation results will be wrong in detectable ways.

5. *The bootstrapping problem*: We are using AI to design systems that will be operated by AI, creating a potential circularity in trust. **Mitigation**: The system is designed to be transparent and auditable. All mechanisms are documented such that humans can verify the logic. The AI does not operate autonomously—it assists human designers who retain judgment and accountability.

The honest position: AI-assisted system design is powerful but not infallible. This paper represents our best current understanding, developed through human-AI collaboration. We expect some aspects to be wrong and invite scrutiny. The simulations, code, and reasoning are published precisely so that others—human and AI alike—can verify, critique, and improve upon this work.

This methodology is itself subject to the freedom-trust tradeoff we describe: by making our process transparent (including AI involvement), we sacrifice some authorial mystique in exchange for verifiability. We believe this tradeoff is worthwhile.

8.9 Limitations

Bootstrap Problem: The network requires initial trusted participants to establish the trust graph. Genesis block contents define starting conditions.

Sophisticated Attackers: Well-resourced attackers willing to invest years in building reputation before striking remain a threat. No system fully prevents long-con attacks.

Parameter Sensitivity: Optimal parameter values require empirical tuning and may vary across network conditions.

Verification Overhead: Random audits impose costs on honest participants. The verification rate must balance security against efficiency.

AI-Assisted Design Uncertainty: As discussed in Section 8.8, AI-assisted analysis carries risks of hallucination and training data limitations. While we have attempted to mitigate these through executable simulations and adversarial review, some errors may remain undetected.

Fundamental Sybil Limits: Douceur [6] proved that Sybil attacks are fundamentally unsolvable without trusted identity verification. Omerta's defenses (age, cluster detection, economic penalties) make Sybil attacks expensive but not impossible. A patient, well-resourced adversary who precreates identities years in advance can eventually attack with mature identities. We mitigate this through continuous behavioral monitoring, but acknowledge the theoretical limitation.

Existing Compute Market Struggles: Decentralized compute platforms built on blockchain (Golem [15], iExec [16], Render Network) have operated for years but struggled with utilization and adoption. This suggests challenges beyond trust mechanisms—possibly network effects, user experience, or fundamental market structure issues. Omerta may face similar challenges regardless of its trust system design. We cannot assume that better trust mechanisms alone will solve adoption problems.

AI Demand Thesis is Speculative: Our argument that machine intelligence creates unbounded compute demand (Section 7.5) is forward-looking and unproven. Current AI demand is large but not literally unbounded. The thesis depends on assumptions about AI capability trajectories that may not hold. If AI development stalls or compute efficiency improves faster than demand grows, the perpetual undersupply assumption fails, and with it the economic model for home provider value creation.

Detection vs. Prevention: As clarified in Section 7.6.1, Omerta detects double-spends but does not prevent them. The system relies on economic penalties making attacks unprofitable, not on making attacks impossible. This is a weaker guarantee than blockchain consensus provides. For applications requiring absolute prevention of double-spending, blockchain remains more appropriate.

9. Conclusion

Trustlessness exists on a spectrum. Proof-of-work and proof-of-stake mechanisms genuinely reduce trust requirements compared to centralized alternatives—this is a real achievement. But they do so at significant cost, and historical episodes demonstrate that no practical system reaches the zero-trust endpoint.

Omerta explores a different point on this spectrum. Rather than paying for global consensus that compute markets may not need, Omerta computes trust locally based on verifiable on-chain data. The hypothesis is that for bilateral compute transactions—where sessions are ephemeral, verification is possible during execution, and reputation has natural meaning—local trust provides sufficient security at dramatically lower cost.

This parallels the choice between fully homomorphic encryption and ephemeral compute. FHE provides the ultimate guarantee but at 1000x+ overhead, restricting practical use. Ephemeral compute accepts some trust requirements in exchange for serving far more use cases. Similarly, blockchain consensus provides strong guarantees but at costs (energy, capital, throughput) that may exceed what compute markets require.

Our simulation studies validate several key claims. Automated policy mechanisms respond appropriately to detected threats, though parameter adjustment alone cannot counter structural attacks—effective systems need architectural defenses complemented by policy fine-tuning. Double-spend resolution simulations confirm that currency "weight" scales with network performance: well-connected networks achieve 100% detection with sub-200ms finality, while poorly-connected networks require heavier mechanisms. The system degrades gracefully across this spectrum.

Our economic simulations demonstrate that unreliable home compute creates genuine value—not merely redistributes it—when demand exceeds datacenter capacity. In undersupplied markets, home providers serve consumers that datacenters cannot reach, increasing total compute delivered by 200% while datacenter profits remain largely intact. The viability of this model depends on perpetual undersupply.

We argue that machine intelligence guarantees this undersupply. Unlike human demand, which is bounded and tends toward equilibrium, machine intelligence creates unbounded demand for compute at any quality level. There is always a next-best task—a machine that cannot profitably do a $50/hr task can still generate value doing a $5/hr task. As AI systems improve, this demand curve extends to arbitrarily low prices, ensuring that any compute capacity finds productive use.

But machine intelligence plays a deeper role than just creating demand. Fair trust systems at scale—systems that penalize only provable misbehavior through transparent, debatable mechanisms—were computationally intractable until now. Modeling human behavior, detecting novel attacks, explaining decisions, and tuning parameters requires enormous reasoning capacity. Machine intelligence provides this capacity for the first time. The relationship is recursive: AI demands the compute that Omerta provides, and AI enables the trust system that makes Omerta work. This virtuous cycle suggests the technologies arrive together because each enables the others.

This framing illuminates what Omerta attempts: extending village-level trust to global scale. Villages had high trust because they had high visibility—everyone knew everyone's business, reputation spread by gossip, misbehavior had lasting consequences. Omerta recreates these properties digitally through on-chain transparency. But unlike villages with their arbitrary social punishment, gossip, and hidden power structures, Omerta aims to maximize freedom within the trust constraint.

Only provably anti-social behavior affects trust scores. All mechanisms are documented and debatable. Participants retain freedom for any behavior that doesn't demonstrably harm others.

The goal is not to replace blockchain systems—they serve real purposes and represent genuine advances. The goal is to expand the design space, recognizing that different applications may have different optimal points on the trust-cost spectrum. For compute markets specifically—especially those serving machine intelligence workloads—we believe there is an underexplored region that could make trustworthy compute sharing accessible to more people at practical cost.

Omerta is an experiment in finding that region. The question is not whether unreliable compute will displace datacenters, but whether we can deploy enough compute of any quality to satisfy the exponentially growing demand of machine intelligence—and whether machine intelligence can, in turn, help us build the fair trust systems needed to make that deployment work.

References

[1] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45-48, 2000.

[2] C. Dellarocas, "The digitization of word of mouth: Promise and challenges of online feedback mechanisms," *Management Science*, vol. 49, no. 10, pp. 1407-1424, 2003.

[3] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in *Proc. WWW*, 2003.

[4] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE TKDE*, vol. 16, no. 7, pp. 843-857, 2004.

[5] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer file-sharing," in *Proc. NSDI*, 2006.

[6] J. R. Douceur, "The Sybil attack," in *Proc. IPTPS*, 2002.

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[8] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," 2012.

[9] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "SybilGuard: Defending against Sybil attacks via social networks," *ACM SIGCOMM*, 2006.

[10] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil nodes using social networks," in *Proc. NDSS*, 2009.

[11] D. Larimer, "Delegated proof-of-stake (DPOS)," *Bitshare whitepaper*, 2014.

[12] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, 1999.

[13] D. P. Anderson, "BOINC: A system for public-resource computing and storage," in *Proc. Grid*, 2004.

[14] V. S. Pande et al., "Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing," *Biopolymers*, vol. 68, no. 1, pp. 91-109, 2003.

[15] The Golem Project, "The Golem whitepaper," 2016.

[16] iExec, "iExec: Blockchain-based decentralized cloud computing," 2017.

[17] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proc. EC*, 2004.

[18] R. Jurca and B. Faltings, "Collusion-resistant, incentive-compatible feedback payments," in *Proc. EC*, 2007.

[19] G. E. Bolton, B. Greiner, and A. Ockenfels, "Engineering trust: Reciprocity in the production of reputation information," *Management Science*, vol. 59, no. 2, pp. 265-285, 2013.

[20] D. E. Denning, "An intrusion-detection model," *IEEE TSE*, vol. 13, no. 2, pp. 222-232, 1987.

[21] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, 2009.

[22] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," in *Proc. CCSW*, 2011.

[23] V. Costan and S. Devadas, "Intel SGX explained," *IACR Cryptology ePrint Archive*, 2016.

[24] J. Van Bulck et al., "Foreshadow: Extracting the keys to the Intel SGX kingdom," in *Proc. USENIX Security*, 2018.

[25] A. C. Yao, "How to generate and exchange secrets," in *Proc. FOCS*, 1986.

[26] L. Tesfatsion and K. L. Judd, Eds., *Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics*. North-Holland, 2006.

[27] J. D. Farmer and D. Foley, "The economy needs agent-based modelling," *Nature*, vol. 460, no. 7256, pp. 685-686, 2009.

[28] W. B. Arthur, "Complexity and the economy," *Science*, vol. 284, no. 5411, pp. 107-109, 1999.

[29] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Eds., *Algorithmic Game Theory*. Cambridge University Press, 2007.

[30] P. Windrum, G. Fagiolo, and A. Moneta, "Empirical validation of agent-based models: Alternatives and prospects," *Journal of Artificial Societies and Social Simulation*, vol. 10, no. 2, 2007.

[31] P. Klemperer, "Auction theory: A guide to the literature," *Journal of Economic Surveys*, vol. 13, no. 3, pp. 227-286, 1999.

[32] A. E. Roth, "The economics of matching: Stability and incentives," *Mathematics of Operations Research*, vol. 7, no. 4, pp. 617-628, 1982.

[33] C. Dellarocas, "Reputation mechanism design in online trading environments with pure moral hazard," *Information Systems Research*, vol. 16, no. 2, pp. 209-230, 2005.

[34] O. E. Williamson, "Transaction cost economics: How it works; where it is headed," *De Economist*, vol. 146, no. 1, pp. 23-58, 1998.

[35] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 13, no. 2, pp. 119-154, 2006.

[36] A. Jøsang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer, 2016.

[37] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460-473, 2007.

[38] E. Budish, "The economic limits of Bitcoin and the blockchain," *Quarterly Journal of Economics*, 2024. (Originally NBER Working Paper 24717, 2018.)

[39] J. S. Gans and N. Gandal, "More (or less) economic limits of the blockchain," *CEPR Discussion Paper*, 2019.

[40] S. Seuken and D. C. Parkes, "On the limitations of reputation systems," in *Proc. EC*, 2014.

[41] M. O. Jackson, *Social and Economic Networks*. Princeton University Press, 2008.

Appendix: Key Parameters

Parameter	Description	Typical Range
K_PAYMENT	Payment curve slope	0.01 - 0.10
K_TRANSFER	Transfer burn slope	0.01 - 0.10
AGE_MATURITY_DAYS	Days to full trust potential	90 - 180
DAILY_MINT	Coins minted per day	Decreasing schedule
ISOLATION_THRESHOLD	Sybil cluster detection	0.7 - 0.9
TRANSITIVITY_DECAY	Trust decay per hop	0.5 - 0.8
DAMPENING_FACTOR	Policy adjustment scaling	0.1 - 0.5